Internal state recovery attack on Stream Ciphers : Breaking BIVIUM

Space-2019
DAIICT, Gandhinagar

Shravani Shahapure[1] Virendra Sule[2] R. D. Daruwala[1]

[1]Department of Electronics,
Veer Jijamata Technological Institute,
Matunga, Mumbai 400031, India

[2]Department of Electrical Engineering,
Indian Institute of Technology Bombay,
Powai, Mumbai 400076, India

December 6,2019

Outline

Main Contributions

- Shows that the implicant based Boolean system solver [6] is effective in solving Boolean equations of Bivium in realistic conditions, time and memory and recover the key from an output stream.
- Shows that the symbolic model of Bivium output streams is computable in practically feasible time and memory.
- Discovered two heuristics for decomposing Boolean equations which are general enough to be useful for cryptanalysis of ciphers whose Boolean models are available offline.

Outline

Main Objectives of the research

- To establish the role and importance of *symbolic modelling* in Boolean cryptanalysis. *Boolean Cryptanalysis* is distinguished from algebraic cryptanalysis by its reliance on Boolean arithmetic computations rather than commutative algebraic arithmetic.
- To emphasise that Boolean cryptanalysis is the most definite general method of Cryptanalysis of lower complexity than brute force search over key space, is parallel and can be used for practical evaluation of security of cryptographic primitives.

Outline

Key Recovery Problem

- General model of a stream cipher as dynamic system is specified by a finite state map $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ giving a state output system

$$\begin{aligned} x(k+1) &= F(x(k)) \\ w(k) &= f(x(k)) \end{aligned} \tag{1}$$

  where $x(k)$ is the internal state (of all the shift registers) and $w(k)$ is the output (key stream bit) of the cipher at an instant k.

- The symmetric key K and IV are part of initial condition x(0).

Problem

Key recovery problem: Given an instant $k_0 > 0$, the IV of the stream cipher and the output stream $w(k_0), w(k_0+1), \ldots, w(k_0+m)$ for some $m > 0$ find the key K.

Outline

Internal state recovery problem

Problem

Internal state recovery problem: Given an instant $k_0 > 0$ and an output stream $w(k)$ for $k = k_0, k_0 + 1, k_0 + 2, \ldots, k_0 + m$ corresponding to an unknown key K and known IV, find the internal state $x(k_0)$.

Internal State Recovery attack:

- Compute all solutions $x(k_0)$ of the internal state from the output stream $w(k)$ for $k \geq k_0$.
- Invert $x(k_0)$ to $x(0)$ by inversion of the state map $F(.)$.
- Identify correct $x(0)$ by matching IV. Recover K from $x(0)$.

Outline

Symbolic model computation

- Output stream:

$$w(k_0 + j) = f_j(x(k_0)) = f(F^{j-1}(x(k_0)) \tag{2}$$

  Offline one time computation of functions in RHS of equation (2). This is the symbolic model of output equations.
- Solution of the Boolean system (2) leads to recovery of internal state $x(k_0)$. There might be several solutions $x(k_0)$.
- Symbolic inverse model $F^{-1}(.) = G(.)$ of the state map required for key recovery. Inverse state update symbolic system

$$x(k-1) = G(x(k))$$

Outline

Solving the Boolean System

- Boolean system given by symbolic model (2)

$$w(k_0 + j) = f_j(x(k_0))$$

- Hardest Computational problem: All solutions of Boolean satisfiability ($\#P$ complete)
- All satisfying assignments required: to get all solutions $x(k_0)$.

Outline

Implicant based Boolean system solver

[6] Virendra Sule. An implicant based parallel all solution solver for Boolean satisfiability. arxiv.org/1611.09590v3. Feb 2017.

- Represents all satisfying assignments of a Boolean System as implicants. (compact representation without free variables).
- Computes a complete set of Orthogonal implicants of the system which gives all assignments as a disjoint (hence parallely computable) union of sets.
- Other SAT algorithms solve the Decision Problem only.
- The algorithm is completely (thread) parallel.
- Does not require CNF form of the Boolean formula. (Conversion to CNF form for complex systems has computational overheads).

Outline

Implicant based Boolean system solver

- A term t is an *implicant* of a Boolean function $f(X)$ in variables $X = (x_1, \ldots, x_n)$ if

$$t(x) \leqslant f(x) \tag{3}$$

  for any assignment x of X.
- A *complete set of implicants* $I(f)$ of f is a set of implicants such that if $f(x) = 1$ for an assignment x then there exists at least one term t in $I(f)$such that $t(x) = 1$.
- Complete implicant set of a product of two functions

$$\begin{aligned} \text{fg:I(fg)} &= \{ts, t \in I(f), s \in I(g/t)\} \\ \text{fg:I(fg)} &= \{ts, t \in I(f/s), s \in I(g)\} \end{aligned} \tag{4}$$

Outline

Implicant Algorithm

- Recursive implicant set generation for a product formula

$$F_m = f_1 f_2 \dots f_m = \prod_{i=1}^{m} f_i$$

- $F_m = f_1 F_{(m-1)}$, $F_{(m-k)} = f_{(k+1)} F_{(m-k-1)}$, $F_1 = f_m$.
- Using (4)

$$I(F_m) = \{ts, s \in I(F_{(m-1)}/t), t \in I(f_1)\}$$

  Recursively the algorithm computes a complete implicant set of $F_{(m-1)}/t$
- Pivote factor f_1 can be chosen arbitrarily.
- Implicants of pivotes chosen Orthogonal leading to orthogonal implicants at each end of thread.

Outline

Heuristics for decomposition of Boolean System

- H1: Search over *top ranked* variables: variables which arise in largest number of functions as arguments. E.g. if x_1 is present in 10 functions f_i and x_2 is present in 11 functions, then x_2 has higher rank than x_1.
- H2: Search over implicants of equations which have non-overlapping variables. The implicant set is the product of implicant sets of individual equations.
- Top ranked variables simplify the equations rapidly.
- Combination of the strategies H1, H2 may be more effective.

Heuristics for decomposition of Boolean System

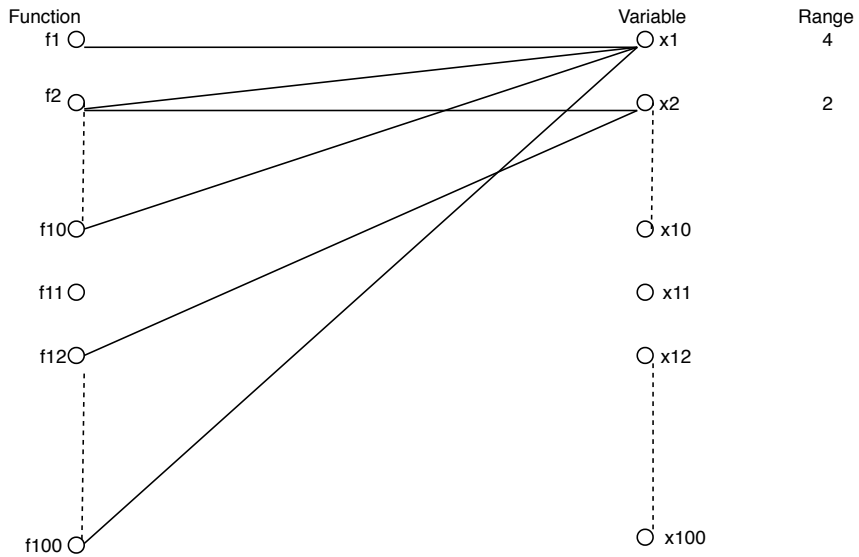

Figure: Variable ranks

Heuristics for decomposition of Boolean System

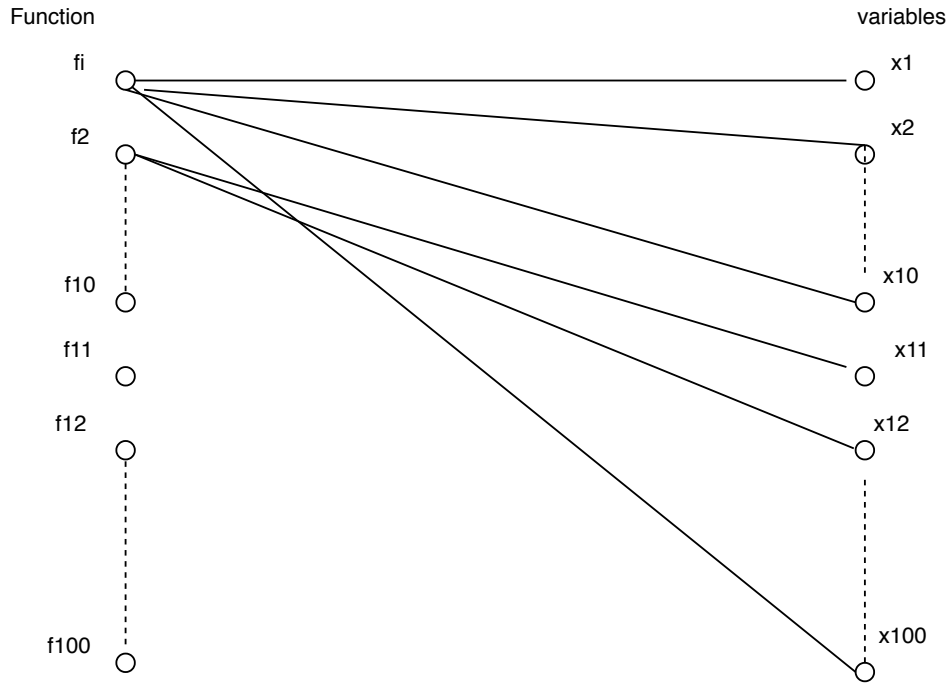

Figure: Functions with non overlapping variables

Outline

Case of Bivium equations

1. First set of 66 equations are linear with 4 variables each. All equations have mutually independent variables (non-overlapping variables in any pair of equations). Hence H2 possible on these equations.
2. 8 implicants in these 4 variables for each equation irrespective of the value of output w. Implicants of different equations have non-overlapping variables.
3. If m of these 66 equations are chosen the search space is $8^m = 2^{3m}$ independent strings of $4m$ variable assignments.

Outline

Results

Table: Implicant based computation timings

No. of eqs	No. of var assgn	No. of impl threads	Avg Time for comp Sec.	Memory required in TB
13	52	2^{39}	$>$ 2 days	3.95
14	56	2^{42}	61503	30.68
15	60	2^{45}	4926	261.16

Outline

Results

Table: Timing with assignment of top ranked variables

No. of top ranked variables in search	Time taken (Sec) to solve the system	Memory required in terabytes
40	177746 (49 Hrs)	5.4
44	100642 (27 Hrs)	96.7
45	50521 (14 Hrs)	197.9
48	6642	> 1000
52	663	> 1000
56	26	> 1000
60	13	> 1000

Outline

Conclusions

- Symbolic model of Bivium cipher is computationally simple and represented with low memory requirements.
- Boolean Cryptanalysis is successful in the case of Bivium in realistic conditions.
- Heuristic H1: Search by assigning top ranked variables. Fast but has higher memory requirement.
- Heuristic H2: constructing search based on equations with non-overlapping variables has lesser memory requirement but higher computational time requirement for the equations in the present case.
- Any cipher with symbolic model computed can be evaluated by the Boolean solver for cryptanalysis. This solver can thus be useful as a universal marker of strength of a cipher.

References

[1] Dibyendu Roy, Pratish Datta, Sourav Mukhopadhyay. Algebraic cryptanalysis of stream ciphers using decomposition of Boolean functions. Journal of Applied Mathematics and Computing. October 2015, Volume 49, Issue 12, pp 397417.
[2] Gregory Bard. Algebraic cryptanalysis. Springer 2009.
[3] Courtois N. ONeil S. Quisquater J. J. Practical algebraic attacks on the Hitag2 stream cipher. Information Security, pp.167176, 2009.
[4] Mihaljevi M. J., Gangopadhyay S., Paul G., Imai H. Internal state recovery of grain-v1 employing normality order of the filter function. Inf. Secur. IET 6(2), pp.5564, 2012b.
[5] McDonald C., Charnes C., Pieprzyk J. Attacking Bivium with Minisat. Technical Report 2007/040, ECRYPT Stream Cipher Project 2007.

[6] Virendra Sule. An implicant based parallel all solution solver for Boolean satisfiability. arxiv.org/1611.09590v3. Feb 2017.
[7] Katti Jayashree, Sule Virendra, Lande B.K. Implicant Based Solver for XOR Boolean Linear Systems, SpringSim-HPC 2017, April 23-26, Virginia Beach, VA, USA, pp.556-567; Society for Modeling and Simulation, ISBN: 1-56555-361-6.

Acknowledgements

Authors gratefully acknowledge support by the project 15DITIR002 under NCETIS at IIT Bombay for the research presented in this paper.

THANK YOU!